The current issue and full text archive of this journal is available on Emerald Insight at:
**www.emeraldinsight.com/0264-4401.htm**

# FPGA-realization of a speed control IC for induction motor drive

Ying-Shieh Kung, Seng-Chi Chen, Jin-Mu Lin and
Tsung-Chun Tseng
*Department of Electrical Engineering,*
*Southern Taiwan University of Science and Technology, Tainan, Taiwan*

Received 31 August 2015
Revised 21 January 2016
Accepted 20 February 2016

## Abstract

**Purpose** – The purpose of this paper is to integrate the function of a speed controller for induction motor (IM) drive, such as the speed PI controller, the current vector controller, the slip speed estimator, the space vector pulse width modulation scheme, the quadrature encoder pulse, and analog to digital converter interface circuit, etc. into one field programmable gate array (FPGA).

**Design/methodology/approach** – First, the mathematical modeling of an IM drive, the field-oriented control algorithm, and PI controller are derived. Second, the very high speed IC hardware description language (VHDL) is adopted to describe the behavior of the algorithms above. Third, based on electronic design automation simulator link, a co-simulation work constructed by ModelSim and Simulink is applied to verify the proposed VHDL code for the speed controller intellectual properties (IP). Finally, the developed VHDL code will be downloaded to the FPGA for further control the IM drive.

**Findings** – In realization aspect, it only needs 5,590 LEs, 196,608 RAM bits, and 14 embedded 9-bit multipliers in FPGA to build up a speed control IP. In computational power aspect, the operation time to complete the computation of the PI controller, the slip speed estimator, the current vector controller are only 0.28 $\mu$s, 0.72 $\mu$s, and 0.96 $\mu$s, respectively.

**Practical implications** – Fast computation in FPGA can speed up the speed response of IM drive system to increase the running performance.

**Originality/value** – This is the first time to realize all the function of a speed controller for IM drive within one FPGA.

**Keywords** Field programmable gate array, Induction motor drive, Modelsim/Simulink co-simulation, Speed control IC

**Paper type** Research paper

## 1. Introduction

An induction motor (IM) is an asynchronous alternating current motor. The IM has many advantages like simplicity, reliability, low cost, and virtually maintenance free. Also, due to good dynamic performance, such as wide speed range, easy to implement, good decouple control, IM becomes a research work in literature (Finch and Giaouris, 2008; Holmes *et al.*, 2012; Novotny and Lipo, 1997; Rodríguez *et al.*, 2014). These applications of IM include electric vehicles, fans, air conditioners, pumps, and some home appliances, etc. Currently, indirect field-oriented control (FOC) technique is one of the choice controllers for high-performance IM drives. The technique known as vector control has resulted in a large change in the field of electrical drives. This is because the IM can be controlled with high performance and provides the same performance as achieved from a separately excited DC motor.

This work was supported by the Ministry of Science and Technology in ROC under Grant No. MOST 104-2221-E-218-012.

Engineering Computations:
International Journal for Computer-
Aided Engineering and Software
Vol. 33 No. 6, 2016
pp. 1835-1852
© Emerald Group Publishing Limited
0264-4401
DOI 10.1108/EC-08-2015-0260

In the realization of the digital controller of IM drives, the PC based, digital signal processor (DSP) based and field programmable gate array (FPGA) based provide a possible system solution in this issue (Curkovic *et al.*, 2013; Sutikno *et al.*, 2013; Soufien *et al.*, 2015; Song *et al.*, 2010). Comparing with FPGA, although the control technique using DSP provides a flexible skill, it suffers from a long period of development and exhausts many resources of the CPU. The advantages of the FPGA includes their programmable hard-wired feature, fast time-to-market, shorter design cycle, embedding processor, hardware software co-design, low power consumption, and higher density; therefore, it has been widely used in the implementation of the digital system and motor drive system (Kung *et al.*, 2009; Monmasson *et al.*, 2011).

Recently, a co-simulation work by electronic design automation (EDA) simulator link has been gradually applied to verify the effectiveness of the Verilog and VHDL code in the motor drive system (Kung *et al.*, 2015; Li *et al.*, 2010). The EDA simulator link (MathWorks, 2016) provides a co-simulation interface between MATLAB or Simulink and HDL simulators-ModelSim (Modeltech, 2012). Using it you can verify a VHDL, Verilog, or mixed-language implementation against your Simulink model or MATLAB algorithm. Therefore, EDA simulator link lets you use MATLAB code and Simulink models as a test bench that generates stimulus for an HDL simulation and analyzes the simulation's response.

In this paper, a co-simulation by EDA simulator link is applied to speed control for IM drive. The IM, inverter and speed command are performed in Simulink and the slip speed estimator and integrator, current vector controller, space vector pulse width modulation (SVPWM) generation, and PI speed controller described by VHDL code are executed in ModelSim. After successful verification in simulation, an FPGA-based experimental system is established for testing the developed VHDL code of speed control intellectual properties (IP) again, and experiment results will validate the effectiveness of the speed control system of IM drive.

## 2. FOC of IM

The mathematical model of a typical IM is described, in two-axis *d*-*q* rotating reference frame, as follows (Novotny and Lipo, 1997):

$$\frac{d}{dt}\begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{\sigma L_s}-\frac{R_r(1-\sigma)}{\sigma L_r} & \omega_e & \frac{L_m R_r}{\sigma L_s L_r^2} & \frac{P\omega_r L_m}{2\sigma L_s L_r} \\ -\omega_e & \frac{-R_s}{\sigma L_s}-\frac{R_r(1-\sigma)}{\sigma L_r} & \frac{-P\omega_r L_m}{2\sigma L_s L_r} & \frac{L_m R_r}{\sigma L_s L_r^2} \\ \frac{L_m R_r}{L_r} & 0 & -\frac{R_r}{L_r} & \omega_e-\frac{P}{2}\,\omega_r \\ 0 & \frac{L_m R_r}{L_r} & -\left(\omega_e-\frac{P}{2}\,\omega_r\right) & -\frac{R_r}{L_r} \end{bmatrix} \begin{bmatrix} i_{ds} \\ i_{qs} \\ \lambda_{dr} \\ \lambda_{qr} \end{bmatrix} + \frac{1}{\sigma L_s}\begin{bmatrix} v_{ds} \\ v_{qs} \\ 0 \\ 0 \end{bmatrix} \quad (1)$$

and the electromagnetic torque can be expressed regarding stator current and rotor flux linkage:

$$T_e = \frac{3P}{4}\frac{L_m}{L_r}\left(i_{qs}\lambda_{dr}-i_{ds}\lambda_{qr}\right) \quad (2)$$

with:

$$\sigma = 1-L_m^2/(L_sL_r) \quad (3)$$

where P is number of poles; $\omega_r$, $\omega_e$, $\omega_{sl}$ are rotor angular speed, electrical angular speed, and slip angular speed, respectively; $v_{ds}$, $v_{qs}$ are d-axis and q-axis stator voltage; $i_{ds}$, $i_{qs}$ are d-axis and q-axis stator current; $\lambda_{dr}$, $\lambda_{qr}$ are d-axis and q-axis rotor flux linkage; $R_r$ is rotor resistance; $L_r$, $L_s$, $L_m$ are rotor inductance, stator inductance, and magnetizing inductance, respectively.

From the third and fourth row in (1), the rotor flux equation can be re-arranged in the form of:

$$\frac{d}{dt}\begin{bmatrix}\lambda_{dr}\\ \lambda_{qr}\end{bmatrix}=\begin{bmatrix}-\frac{R_r}{L_r} & \omega_{sl}\\ -\omega_{sl} & -\frac{R_r}{L_r}\end{bmatrix}\begin{bmatrix}\lambda_{dr}\\ \lambda_{qr}\end{bmatrix}+\frac{L_mR_r}{L_r}\begin{bmatrix}i_{ds}\\ i_{qs}\end{bmatrix} \tag{4}$$

where $\omega_{sl}$ is the speed slip defined by $\omega_e-(P/2)\omega_r$.

In the ideally de-coupled IM using FOC, the rotor flux linkage is forced to align with the d-axis, and it follows that:

$$\lambda_{qr}=0 \text{ and } \frac{d}{dt}\lambda_{qr}=0 \tag{5}$$

Therefore, from the first row in (4), the desired rotor flux linkage can be in term of $i_{ds}$ as:

$$\lambda_{dr}=\frac{L_mi_{ds}}{1+L_rs/R_r} \tag{6}$$

Comparing with the mechanical system, the dynamic term in (6) can be neglected, and the equation can be represented by:

$$\lambda_{dr}^*=L_mi_{ds} \tag{7}$$

Therefore, substituting the (5) and (7) into (2), the generated torque becomes:

$$T_e=K_ti_{qs}^* \tag{8}$$

with:

$$K_t=\frac{3P}{4}\frac{L_m^2}{L_r}i_{ds}^* \tag{9}$$

In (8), it shows that the electromagnetic torque is proportional with the current of $i_{qs}^*$. Further, substituting (5) into the second row of (4), the estimated slip speed can be given as follows:

$$\hat{\omega}_{sl}=\frac{R_ri_{qs}^*}{L_ri_{ds}^*} \tag{10}$$

Finally, the IM dynamic equation with mechanical load is given by:

$$J_m\frac{d}{dt}\omega_r+B_m\omega_r=T_e-T_L \tag{11}$$

where $T_e$ is the motor torque, $K_t$ is force constant; $J_m$ is the total inertial value; $B_m$ is total damping ratio; $T_L$ is the external torque.

According to the FOC approach for IM, the architecture of an FPGA-realization speed control system for IM drive is shown in Figure 1. Inside the FPGA, it shows a Nios II processor for speed command generation, others it displays the configuration of a speed loop PI controller and a current control loop using FOC. In the current control circuit, it includes two PI controllers, one coordinate transformations of Clark, modified inverse Clark, Park, inverse Park as well as a SVPWM, a slip speed estimation and integrator, a quadrature encoder pulse (QEP) interface circuit and other element. Except FPGA, the system in Figure 1 also has an inverter, a rectifier, two analog-to-digital converters (ADCs) and one IM, etc. The formulations about the transformations among the stationary *a-b-c* frame, the stationary *α-β* frame and the synchronously rotating *d-q* frame in Figure 1. Finally, the difference form of a digital PI controller in the speed loop and the current loop can be expressed as follows:

$$u_p(n) = K_p e(n) \tag{12}$$

$$u_i(n) = u_i(n-1) + K_i e(n-1) \tag{13}$$

$$u(n) = u_p(n) + u_i(n) = u_i(n-1) + K_p e(n) + K_i e(n-1) \tag{14}$$

where the $u_p$, $u_i$, $u$ are the output of *P* controller, *I* controller and *PI* controller, respectively, as well as the $K_p$, $K_i$ are the gain of the *P* controller and *I* controller, respectively.

## 3. The design of the FPGA-based speed control IC for IM drive

Figure 2 illustrates the internal architecture of the proposed FPGA implementation of a speed control IC for IM drive system. The FPGA herein uses Cyclone IV-EP4CE115, which is the product of Altera cooperation. There are 114,480 logic elements (LEs), 3,981,312 RAM bits, 532 embedded 9-bit multipliers and maximum 529 available I/O pins in the Cyclone IV. The internal circuit comprises a Nios II embedded processor IP and a speed control IP. The Nios II processor is depicted to generate the speed command and collect the response data. All programs in Nios II processor are coded in the C language. The speed control IP includes mainly a circuit of an a speed PI controller, a slip speed estimator and an integrator, a circuit for current controllers and coordinate transformation (CCCT), an SVPWM circuit, a QEP interface circuit and an ADC interface circuit. The sampling frequency of the speed control loop and current control loop are designed with 2 and 16 kHz, respectively. The frequency divider generates 50 MHz (Clk) and 12.5 MHz (Clk-step) clock to supply all circuits in Figure 2.

Herein, only the digital hardware circuit design of the CCCT, the slip speed estimator and integrator, and the PI controller are described and shown in Figures 3-5. To reduce the FPGA resource usage, a finite state machine (FSM) is employed to model the algorithm as mentioned earlier. The design circuits in Figures 3-5 that uses one adder, one multiplier, one divider, some look-up tables, comparators, shifters, registers, etc. to carry out the overall computation. The multiplier, adder, and divider apply Altera library parameterized modules (LPM) standard. The internal circuit of CCCT performs the function of two PI controllers, table look-up for sin/cos function

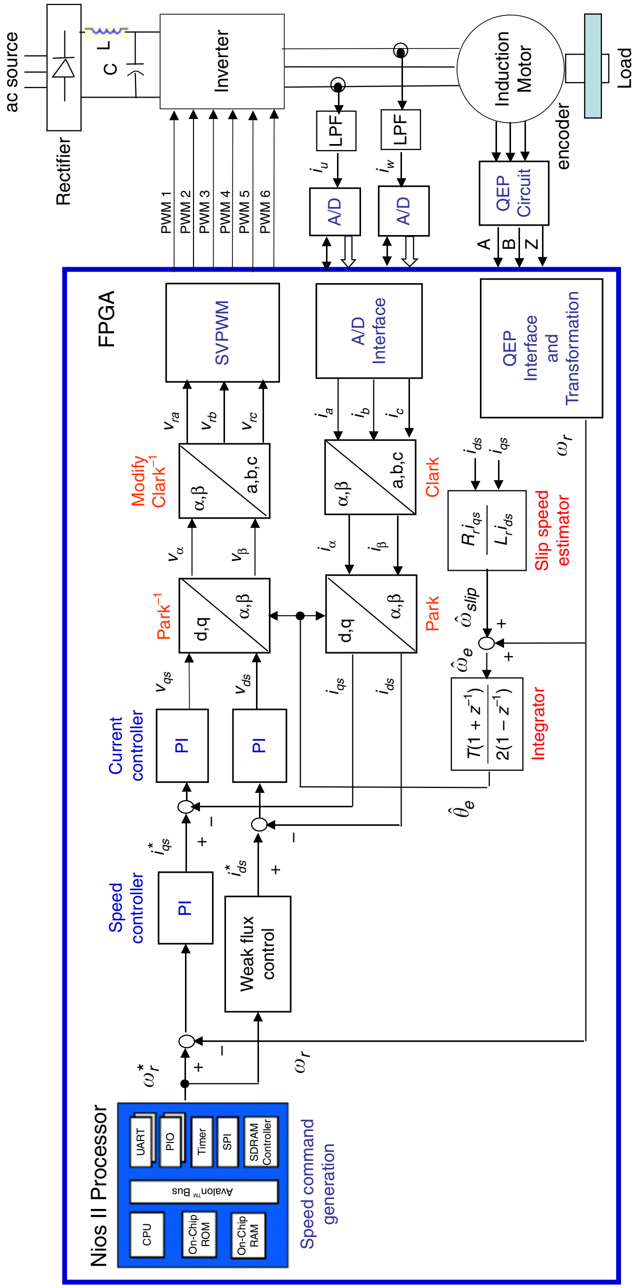


**Figure 1.**
FPGA-realization of a FOC-based speed control system for IM drive

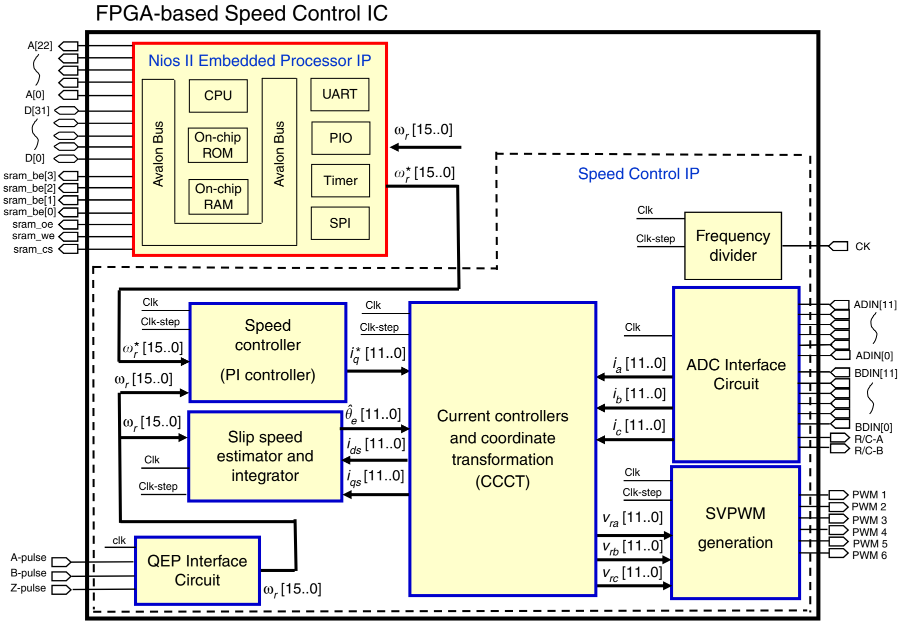


**Figure 2.**
Internal architecture of an FPGA-based speed control IC for IM drive

and the coordinate transformation for Clark, Park, inverse Park, modified inverse Clarke. The CCCT circuit designed using by FSM is shown in Figure 3 and it manipulates 24 steps machine to carry out the overall computation. The data type is 12-bit length with a Q11 format and 2's complement operation. In Figure 3, steps $s_0$~$s_1$ is for the look-up sin/cos table; steps $s_2$~$s_5$ and $s_5$~$s_8$ are for the transformation of Clarke and Park, respectively; steps $s_9$~$s_{14}$ is for the computation of d-axis and q-axis PI controller; and steps $s_{15}$~$s_{19}$ and $s_{20}$~$s_{23}$ represent the transformation of the inverse Park and the modified inverse Clarke, respectively. The operation of each step in FPGA completes within 40 ns (25 MHz clock); therefore whole 24 steps need 0.96 $\mu$s operation time. Although the FSM method requires more operation time than the parallel processing method in executing CCCT circuit, it does not lose any control performance in the overall system because the 0.96 $\mu$s operation time is much less than the designed sampling interval, 62.5 $\mu$s (16 kHz) of the current control loop in Figure 1. Further, the slip speed estimator and integrator developed using by FSM are shown in Figure 4, and it manipulates 18 steps machine to carry out the overall computation. In Figure 4, steps $s_0$~$s_8$ performs the function of the slip speed estimation; steps $s_9$~$s_{10}$ is the summation of slip speed and rotor speed; steps $s_{11}$~$s_{13}$ executes the integration and $s_{14}$~$s_{17}$ generates the LUT address for rotor position, respectively. In the division operation in Figure 4, the dividend, divisor, and quotient are 32-bit Q0, 16-bit Q0, and 16-bit Q8 format, respectively. The divider is a component in the LPM provided by Altera cooperation. Also, the transfer function for the trapezoid rule integration in Figure 1 is shown as:

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{T_{samp}}{2}\frac{1+z^{-1}}{1-z^{-1}} \tag{15}$$

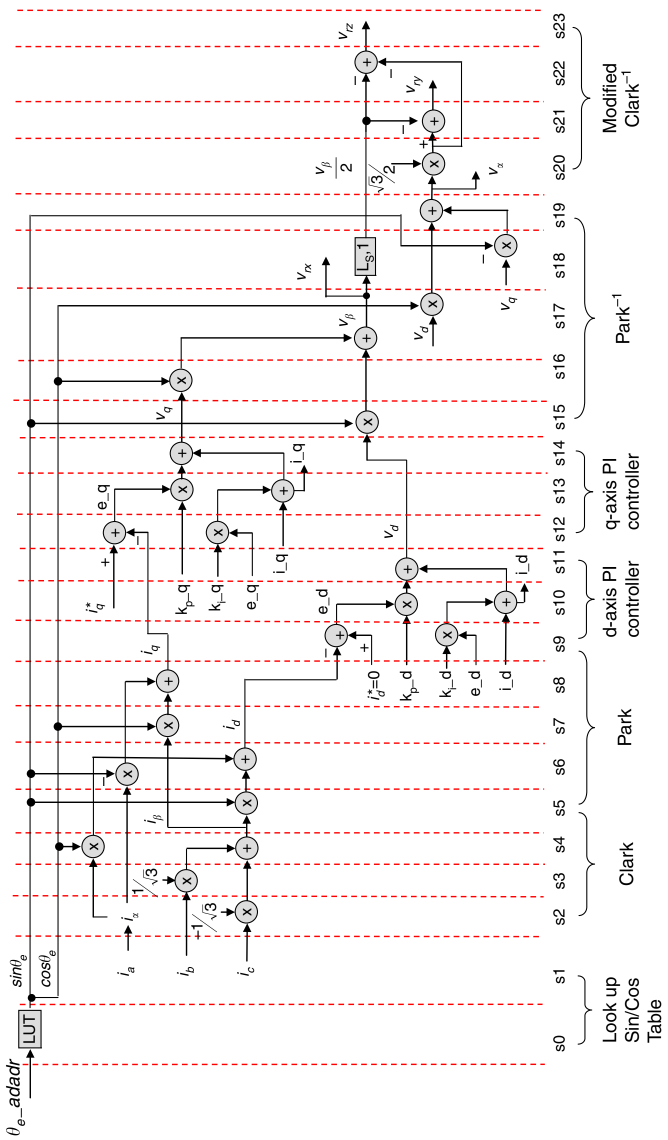


**Figure 3.**
Designed CCCT
circuit using FSM

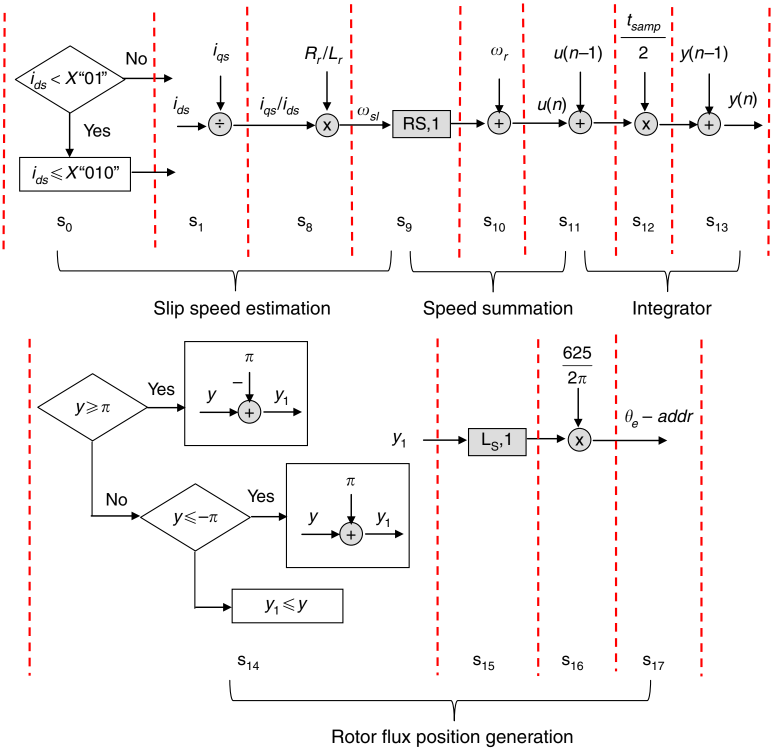


**Figure 4.**
Designed slip speed estimator and integrator using FSM

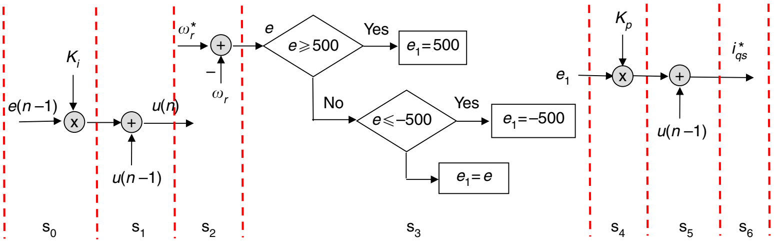


**Figure 5.**
Designed PI controller using FSM

and it will be converted to the difference equation for realization. It is presented as follows:

$$y(n) = y(n-1) + \frac{T_{samp}}{2}(u(n) - u(n-1)) \tag{16}$$

where $T_{samp}$ is the sampling time. Further, the data type in the slip speed estimator and integrator uses 16bit Q6 format and 32bit Q22 format, respectively. The ratio of $R_r/L_r$ will be calculated off-line and taken as a constant input value. In Figure 4, total 18 steps need 0.72 $\mu$s operation time. Finally, the PI controller designed using by FSM is shown in Figure 5 and it manipulates seven steps machine to carry out the overall computation. In the PI controller design, the speed error adopts the 16bit Q0 format. The value will multiply a gain with 64; then be converted to a 16bit Q15 format for further executing the calculation of the PI controller. The range of a singed 16bit Q0 number is within ±32,768, so after it divides by 64, the speed error is limited within ±500 rpm. In Figure 5, whole seven steps need 0.28 $\mu$s operation time.

Finally, the FPGA utility of the speed control IC for IM drive in Figure 2 is evaluated and the result is listed in Table I. The resource usage of the controller architecture shows that the overall circuits of the proposed speed control IC. It includes a Nios II embedded processor IP (8,701 LEs, 28,842 RAM bits and 12 embedded 9-bit multipliers) and a speed control IP (5,590 LEs, 196,608 RAM bits and 14 embedded 9-bit multipliers), which uses 12.48 per cent of the LEs resources, 5.66 per cent of the RAM resources and 2.63 per cent of the embedded 9-bit multipliers resources of a Cyclone IV-EP4CE115.

## 4. ModelSim/Simulink co-simulation of speed control IC for IM

The Simulink/Modelsim co-simulation architecture for speed control system for IM drive is shown in Figure 6. The SimPowerSystem blockset in the Simulink executes the IM and the IGBT-based inverter. The EDA simulator link for ModelSim performs the co-simulation using VHDL code running in ModelSim program. It implements the function of the speed controller by three works. The work-1 to work-3 of ModelSim in Figure 6 performs the function of speed loop PI controller, the function of CCCT and SVPWM, and the function of slip speed estimator and integrator, respectively. The sampling frequency in work-2 and work-3 is designed with 16 kHz but in work-1 is 1 kHz. The clocks of 50 and 12.5 MHz will supply all works of ModelSim. The PI parameters in work-1 and work-2 can be set in the Simulink. In the simulation environment in Figure 6, the speed command is generated by the "speed cmd" block. Further, the speed command and the rotor speed feedback from the "Induction Motor" block will be sent into work-1 to perform the calculation of the speed loop PI controller and generate an output of the q-axis current command to work-2. Except $i_q$-axis current command, the three-phase current detected from the "Induction Motor" block and the rotor position (or electrical θ) estimated from the work-3 will be sent into the work-2 to execute the function of CCCT and SVPWM; then generate the six-channel PWM outputs to drive the IGBT-based inverter working and the IM running. Also, the work-3 will receive the $i_q$ current and $i_d$ current from the work-2 and the rotor speed feedback from the "Induction Motor" block to perform the estimation of the slip speed and the rotor position (or electrical θ). Several scopes in Figure 6, such as named with "rotor speed," "speed slip," "address," "$\mathrm{i_q}$-$\mathrm{i_q}$*," "$\mathrm{i_d}$-$\mathrm{i_d}$*," etc. will, respectively display their simulation results throughout the overall simulation time.

| IP | Module circuit | Logic elements (LEs) | Memory (bits) | Embedded 9-bit multiplier |
|---|---|---|---|---|
| *Nios II embedded processor IP* | | 8,701 | 28,824 | 12 |
| Speed control IP | PI controller | 588 | 0 | 2 |
| | Slip speed estimator and integrator | 2,575 | 0 | 10 |
| | Current controllers and coordination transformation (CCCT) | 772 | 196,608 | 2 |
| | SVPWM generation | 1,324 | 0 | 0 |
| | ADC interface circuit | 180 | 0 | 0 |
| | QEP interface circuit | 151 | 0 | 0 |
| Total | | 14,291 | 225,432 | 26 |

**Table I.** The FPGA utility evaluation of the proposed speed control IC in Figure 2

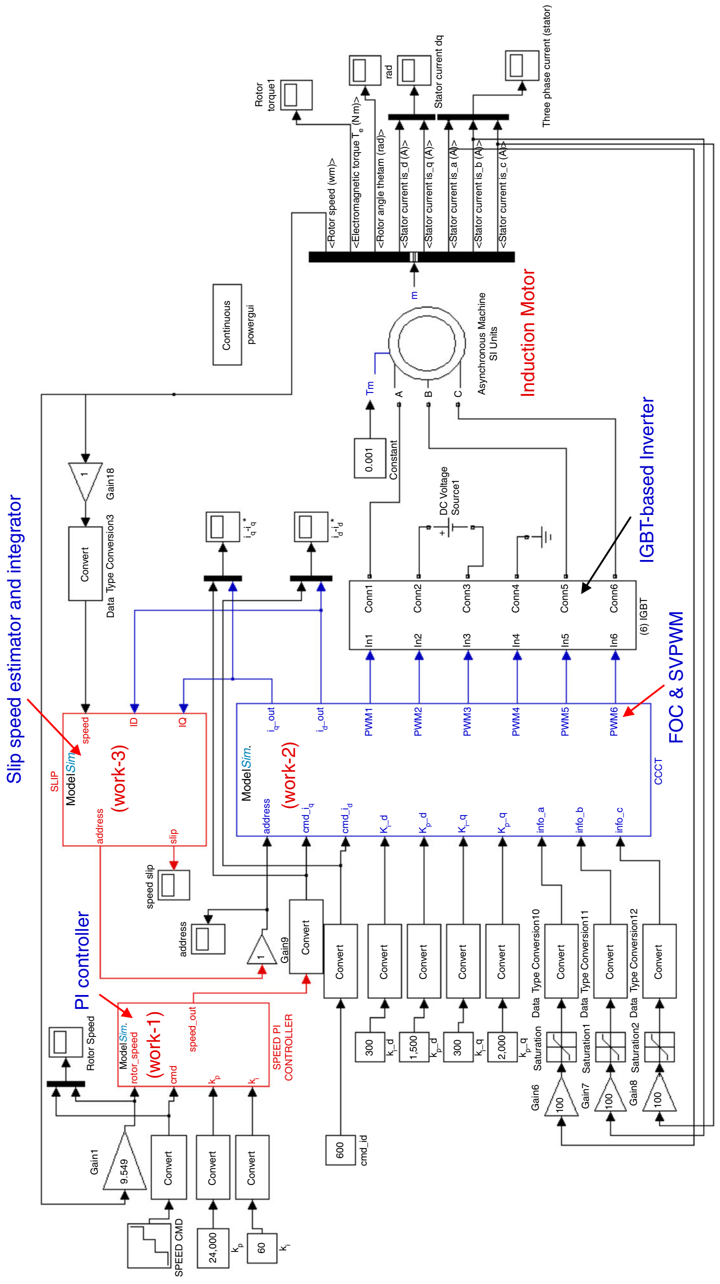
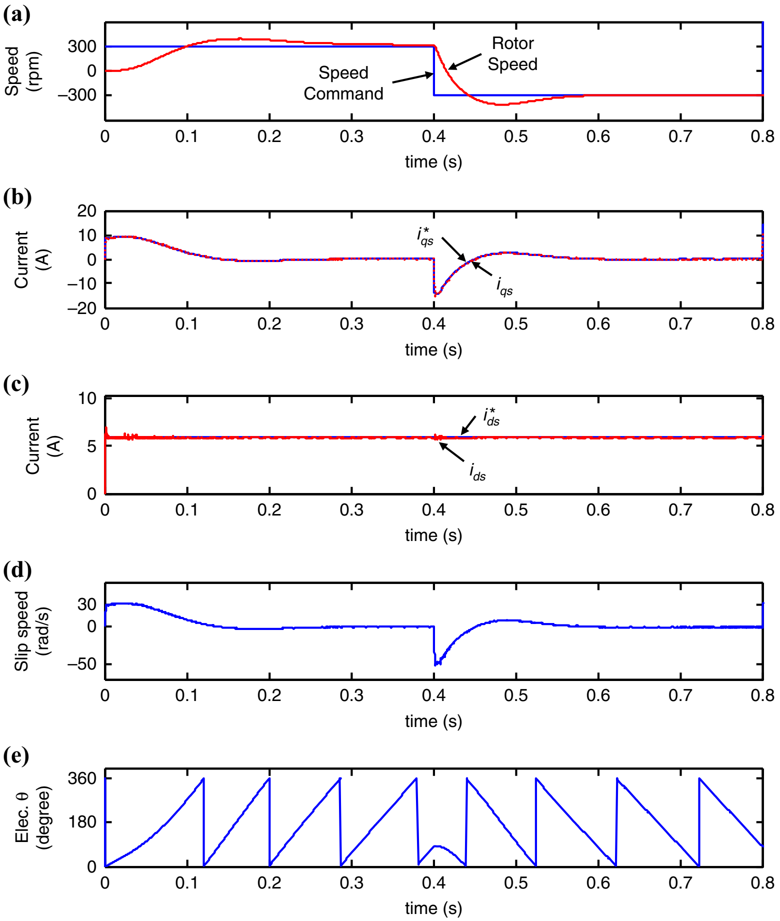


**Figure 6.** The Simulink/ ModelSim co-simulation architecture for speed control of IM drive

In the simulation, the designed PMSM parameters used in Figure 6 are that pole pairs is 2, stator resistance $R_s$ is 1.115 $\Omega$; stator inductance $L_s$ is 5.974 mH; rotor resistance $R_r$ is 1.083 $\Omega$; rotor inductance $L_r$ is 5.974 mH; mutual inductance $L_m$ is 203.7 mH; inertia is $J = 0.02$ kg m$^2$; friction factor is $F = 0.005752$ N m s; base voltage is 460 V (rms); base frequency is 60 Hz; rated power is 3.730 kW (5HP). In current loop, the PI gains in d-axis and q-axis are chosen by $K_p = 1{,}500$ (12bit Q11), $K_i = 300$ (12bit Q11) and $K_p = 2{,}000$ (12bit Q11), $K_i = 300$ (12bit Q11). In speed loop, the PI gains are set by $K_p = 24{,}000$ (16bit Q15) and $K_i = 60$ (16bit Q15). In the proposed system, the maximum current is limited by 20 A and the current for 20.4 A is equivalent with 2,047 by 12bit Q11 format. The external load is 0.001 N m.

To evaluate the proposed controller performance, IM running command at low speed (300 rpm) and inverse speed (from 300 to −300 rpm or vice versa) condition is first considered. Also, its simulation results regarding as the speed step response, the d-axis and q-axis current response, the slip speed estimation and the electrical angle are shown in Figure 7. It shows that the rotor speed in Figure 7(a) presents a dynamic



**Notes:** (a) Speed step response; (b) q-axis current response; (c) d-axis current response; (d) slip speed estimation; (e) electrical θ response

**Figure 7.** Simulated results for IM running at +300 rpm to −300 rpm condition

response performance with a 90 ms rising time, 25 per cent overshoot and near 0 steady-state error. It also demonstrates a successful FOC in Figure 7(b) and (c) that the actual current can track the current command very well both in d-axis and q-axis direction. Figure 7(d) and (e) also show the results of the slip speed estimation and electrical angle response. The maximum/minimum slip speed in transient condition is about 32 rad/s (305 rpm) and −50 rad/s (about −477 rpm), but the slip speed value is near 0 value in the steady-state condition. Further, another simulation case while the IM runs at high speed (from 0 → 300 → 600 → 900 rpm) is tested, and its simulation results are shown in Figure 8. The rotor speed response shown in Figure 8(a) reveals a dynamic response performance with a 90 ms rising time in the initial step speed command from 0 to 300 rpm, but only 40 ms rising time in the following step speed command from 300 to 600 rpm and from 600 to 900 rpm. It also shows that the overshoot is about 25 per cent, and the steady-state error is near 0. The current responses in Figure 8(b) and (c) are similar

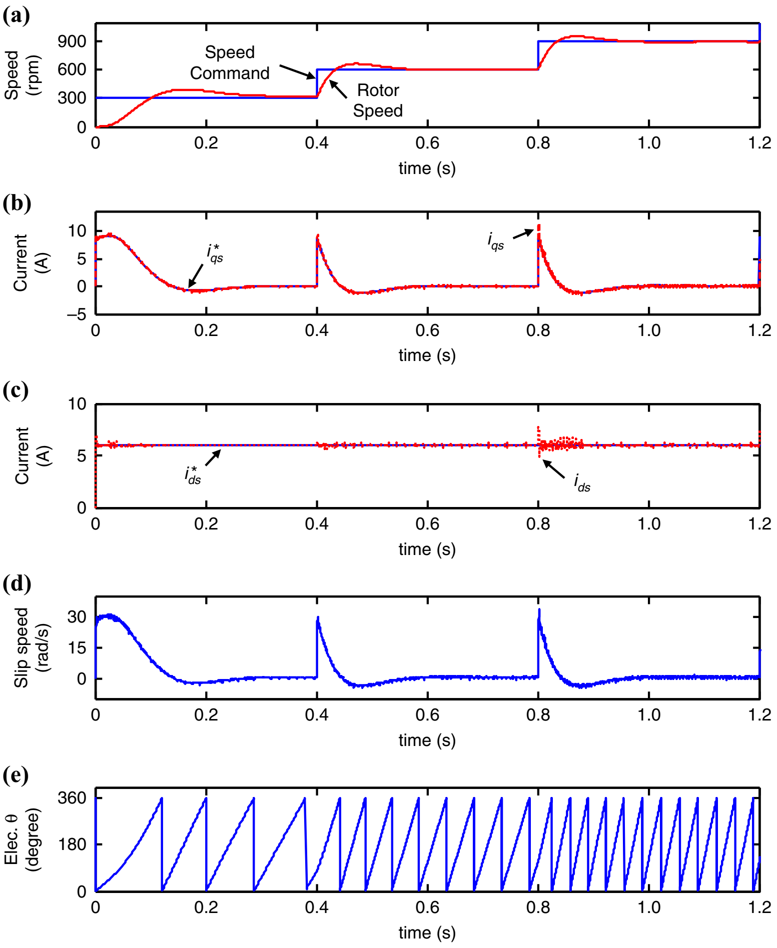


**Figure 8.**
Simulated results for IM running at 300 to 900 rpm condition

**Notes:** (a) Speed step response; (b) q-axis current response; (c) d-axis current response; (d) slip speed estimation; (e) electrical θ response with those in Figure 7(b) and (c). The slip speed response curve in Figure 8(d) is proportional to the q-axis current response that exhibit that a successful FOC is achieved in the current loop system. However, the maximum slip speed in the transient condition is about 35 rad/s (333 rpm), but in the steady-state condition is near 0. The simulated results displayed in Figures 7 and 8 demonstrate the correctness and effectiveness of the proposed speed control IP for IM drive.

## 5. The experimental system and results of the FPGA-based speed control IC for IM

After confirming the effectiveness of the proposed speed control IP by ModelSim/Simulink co-simulation, the VHDL codes for speed control IP are directly applied to the experimental FPGA-based IM drive system. The experimental system is presented in Figure 9. The main devices are an IM, a DE2-115 board with Altera Cyclone IV-EP4CE115 FPGA, a voltage source intelligent power modules-based (IPM) inverter and a rectifier. The input voltage, continuous current, rating torque, rating speed, continuous power, and pole pairs of the IM are 220 V, 5.6 A, 6.4 N m, 1,800 rpm, 1,500 W, 2, respectively. An encoder with 1,024 pulses/rotate is attached at the IM. Inside IPM, it has six sets of IGBT type power transistors with emitter of the rated voltage of 600 V, collector DC rating of 20 A, and a short time (1 ms) rated current of 40 A. Photo coupler IC adopts TLP250, which output has push-pull amplifier functions. Input signals of the inverter are PWM signals from the FPGA device. The Altera Cyclone IV-EP4CE115 chip adopted in the design possesses 114,480 LEs, maximum 529 available I/O pins, 3,981,312 RAM, and 532 embedded 9-bit multipliers. The chip can embeds with a Nios II processor that is equipped with several 32-bit CPU, a flexibility of core size, 1-16 Mbytes of flash memory in the available memory chip, 1 Mbyte SRAM, 16 Mbyte SDRAM, and 4 Gbytes memory outside of the chip.

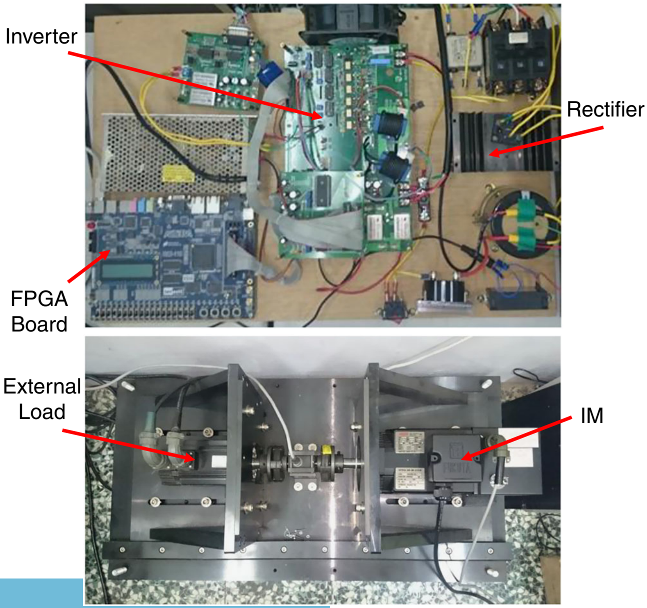


**Figure 9.** Experimental system

In the implementation, the VHDL codes of speed control IP which includes CCCT, PI controller, slip speed estimator and integrator, ADC and QEP interface circuits, will integrate with Nios II embedded processor IP, to downloaded into FPGA. In the controller design, the PI gains in d-axis and q-axis are chosen by $K_p = 800$ (12bit Q11), $K_i = 10$ (12bit Q11) in the current loop. In the speed loop, the PI gains are set by $K_p = 24{,}000$ (16bit Q15) and $K_i = 60$ (16bit Q15). In the flux control, the d-axis current command is set at 1.5 A. In the realization, two control approaches are adopted to IM drive. One is the scalar control which typically applies in the industry; the other is the FOC. The former approach is based on the voltage-frequency (V/f) control method and shown in Figure 10, and the latter approach is presented in Figure 1. The scalar controller shown in Figure 10 is an open-loop control structure with no current feedback but closed-loop control architecture in the speed loop system. The V/f curve in Figure 10 is 220/60 V/Hz.

In the experiment, the control performance of the scalar control approach is first evaluated. When the speed command of the IM drive is a square wave with a 3.3 s period, and with two different magnitude ranges from 900-1,200 rpm and 600-900 rpm, the speed step responses are reveals in Figure 11(a) and (b), respectively. However, the transient conditions present very fast response, but the steady-state conditions appear severe oscillation. Second, the FOC approach applied to IM drive in Figure 1 is evaluated. When the speed command runs at the condition that the speed command is a square wave with magnitude ±300 rpm and period 2 s is considered. The experiment results regarding as the speed step response, the d-axis and q-axis current response and slip speed estimation are shown in Figure 12. In Figure 12(a), it shows that the motor speed gives a good dynamic response performance with a little overshoot and near 0 steady-state error. It has 20 ms rising time. Figure 12(b) shows the actual d-axis current can track to the d-axis current command at 1.5 A and the actual q-axis current can follow to the q-axis current command very well. It reveals that the de-coupled effect is a success after the FOC is adopted. Figure 12(c) shows the slip speed response and its maximum/minimum slip speed within ± 330 rpm. Further, the IM running at a wide-range speed from 300 →600→ 900→ 1,200→ 1,500 rpm is tested, and the experiment results regarding as the speed step response, the d-axis and q-axis current response and slip speed estimation are shown in Figure 13. In Figure 13(a), the rotor speed can track the speed command well with 0 error in the steady-state condition as well as 5~10 per cent overshoot and about 20 ms rising time in the transient condition. In Figure 13 (b), the FOC also present the success due to the actual d-axis and q-axis current that can track the current command very well. In Figure 13(c), it shows the slip speed estimation with maximum 460 rpm. The result is reasonable because the slip speed response is similar to the response in q-axis current. However, comparing with the results of Figures 11-13, the speed control performance of IM drive using the FOC approach is more smooth and accurate than the scalar control approach. Finally, the power consumption is also evaluated. When the voltage of the DC bus provides a 220 V, the current and power use in different operating speed (300-1,800 rpm) of IM drive using two different control approaches is listed in Table II. It presents that the current value measured in DC bus are between 0.25 and 0.46 A if the scalar control approach is applied, but between 0.06 and 0.16 A if the FOC approach is used. Besides, the saving power rate is about 65-75 per cent when the FOC approach is adopted.

Therefore, from the simulation results in Figures 7-8 and the experimental results in Figures 11-13, it demonstrates that the proposed FPGA-based speed control IC used in IM drive is effectiveness and correctness.

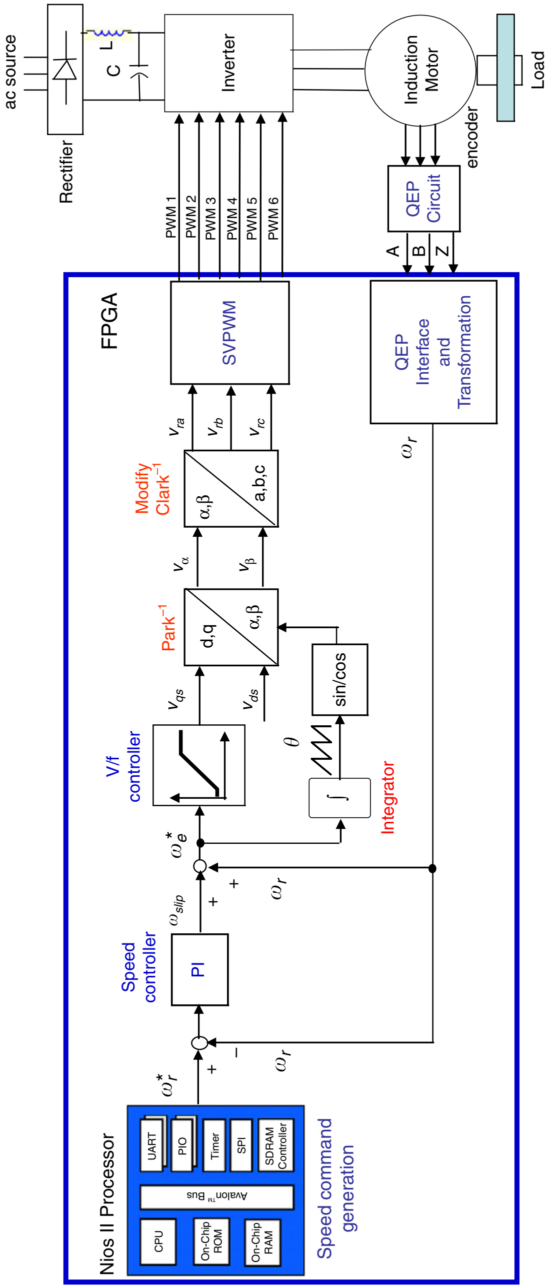


**Figure 10.**
The speed control system for an IM drive based on the scalar control approach

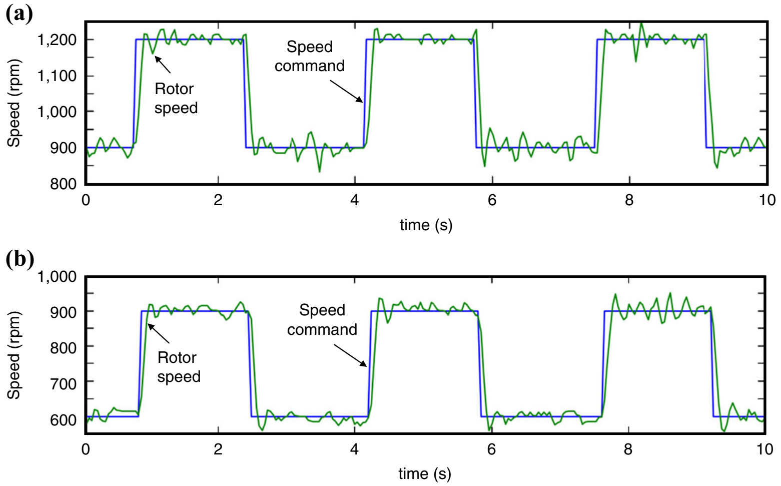


**Figure 11.** Speed step response of IM running at (a) 900-1,200 rpm and (b) 600-900 rpm condition when the controller adopts scalar control approach

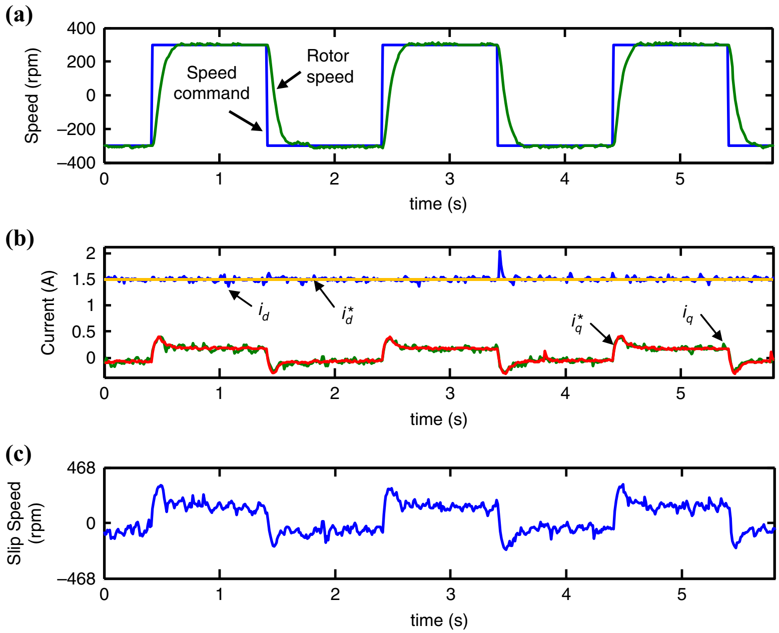


**Figure 12.** IM running at ±300 rpm condition and its (a) speed step response (b) current response (c) slip speed estimation when the controller adopts FOC approach

## 6. Conclusion

In this paper, an FPGA-based speed control IC for IM drive has been successfully demonstrated its performance through co-simulation by using Simulink/ModelSim and implementation by using FPGA. In realization aspect, the VHDL is used to describe the behavior of the PI controller, the slip speed estimator, the CCCT, the SVPWM generation, the QEP, and ADC interface circuit. The FSM is adopted for reducing the FPGA resource usage; therefore, it only needs 5,590 LEs, 196,608 RAM bits and 14 embedded 9-bit multipliers to build up the speed control IP. In computational power aspect, the operation

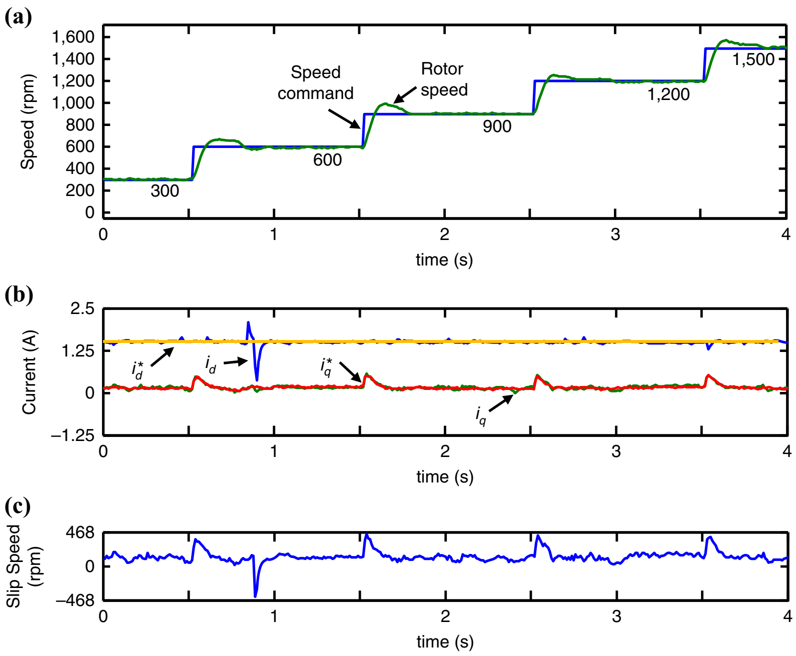


**Figure 13.** IM running at the condition from 300 rpm to 1,500 rpm and its (a) speed step response (b) current response (c) slip speed estimation when the controller adopts FOC approach

| Operating speed of IM | Current/power consumption<br>Scalar approach | <br>FOC approach | Saving power rate (%) |
|---|---|---|---|
| 300 rpm | 0.25 A/55.0 W | 0.06 A/13.2 W | 76.0 |
| 600 rpm | 0.27 A/59.4 W | 0.08 A/17.6 W | 70.3 |
| 900 rpm | 0.32 A/70.4 W | 0.10 A/22.0 W | 68.7 |
| 1,200 rpm | 0.36 A/79.2 W | 0.12 A/26.4 W | 66.6 |
| 1,500 rpm | 0.40 A/88.0 W | 0.14 A/30.8 W | 65.0 |
| 1,800 rpm | 0.46 A/101.2 W | 0.16 A/35.2 W | 65.2 |

**Note:** DC bus voltage: 220 V

**Table II.** Power consumption evaluation under two different control approaches

time to complete the computation of the PI controller, the slip speed estimator, the CCCT are, respectively only 0.28, 0.72, and 0.96 $\mu$s, which are less than the 62.5 $\mu$s (16 KHz) sampling time in the current control loop. In the experimental results, it shows that the FOC approach reveals a better control performance and saving power than the scalar control approach. However, it also demonstrates that the proposed FPGA-based speed control IC is effectiveness and correctness both from the simulated and experimental results.

## References

Curkovic, M., Jezernik, K. and Horvat, R. (2013), "FPGA-based predictive sliding mode controller of a three-phase inverter", *IEEE Transations on Industrial Electronics*, Vol. 60 No. 2, pp. 637-644.

Finch, J.W. and Giaouris, D. (2008), "Controlled AC electrical drives", *IEEE Transations on Industrial Electronics*, Vol. 55 No. 2, pp. 481-491.

Holmes, D.G., Mcgrath, B.P. and Parker, S.G. (2012), "Current regulation strategies for vector-controlled induction motor drives", *IEEE Transations on Industrial Electronics*, Vol. 55 No. 2, pp. 481-491.

Kung, Y.S., Fung, R.F. and Tai, T.Y. (2009), "Realization of a motion control IC for X-Y table based on novel FPGA technology", *IEEE Transactions on Industrial Electronics*, Vol. 56 No. 1, pp. 43-53.

Kung, Y.S., Nguyen, P.T. and Wang, M.S. (2015), "Design and simulation of a sensorless PMSM drive with microprocessor-based PI controller and dedicated hardware EKF estimator", *Applied Mathematical Modelling*, Vol. 39 No. 19, pp. 5816-5827.

Li, Y., Huo, J., Li, X., Wen, J., Wang, Y. and Shan, B. (2010), "An open-loop sin microstepping driver based on FPGA and the co-simulation of Modelsim and Simulink", *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering*, pp. 223-227.

MathWorks (2016), *Simulink Getting Started Guide*, The MathWorks Inc., Massachussetts, MA.

Modeltech (2012), *ModelSim User's Manual*, Mentor Graphics Corporation, Oregon.

Monmasson, E., Idkhajine, L., Cirstea, M.N., Bahri, I., Tisan, A. and Naouar, M.W. (2011), "FPGAs in industrial control applications", *IEEE Transations on Industrial Informatics*, Vol. 7 No. 2, pp. 224-243.

Novotny, D.W. and Lipo, T.A. (1997), *Vector Control and Dynamics of AC Drives*, Oxford University Press, New York, NY.

Rodríguez, R., Gómez, R.A. and Rodríguez, J. (2014), "Fast square root calculation for DTC magnetic flux estimator", *IEEE Latin America Transactions*, Vol. 12 No. 2, pp. 112-115.

Song, Y., Ma, J., Zhang, H. and He, N. (2010), "Digital implementation of neural network inverse control for induction motor based on DSP", *Proceedings of the International Conference on Future Computer and Communication (ICFCC), Vol. 1*, pp. V1-174-V1-178.

Soufien, G., Abdellatif, M. and Mohamed, F.M. (2015), "Design and experimental implementation of DTC of an induction machine based on fuzzy logic control on FPGA", *IEEE Transations on Fuzzy Systems*, Vol. 23 No. 3, pp. 644-655.

Sutikno, T., Idris, N.R., Jidin, A. and Cirstea, M.N. (2013), "An improved FPGA implementation of direct torque control for induction machines", *IEEE Transations on Industrial Informatics*, Vol. 9 No. 3, pp. 1280-1290.

**Corresponding author**
Ying-Shieh Kung can be contacted at: kung@stust.edu.tw